Vessel Container Format

INTERPEER

@IETF115 // ICNRG

Introduction

Interpeer Project does R&D in “internet technology”;

This talk is one building block in an ICN (-like?) system.

Work done under a grant from NLNet/NGI Zero, but influenced by findings from work for ISOC foundation.

https://nlnet.nl/

https://www.ngi.eu/ngi-projects/ngi-zero/

https://www.isocfoundation.org/

Goals

Streaming friendly;

- Multiplexing for e.g. video, audio, subtitles;

Offline-first usage supported (eventual consistency);

Multiple authors/contributors to a single resources;

- Collaborative editing requires something like a CRDT;

E2EE by default (see DINRG talk);

GDPR/right to be forgotten: deletion of content;

Vessel and ICN

Concerned with Named Data Objects (RFC7927): “object name” is not just a hash;

Provides data origin authentication (Section 4.1);

Has partial solution for distribution of public key information (Section 4.2.1) by embedding such data;

E2EE and this embedding also addresses Sections 4.2.3 and 4.2.4.

Alternative Approaches

Using a hash as an object name is elegant and robust, but is very final – no editing or deletion possible.

FLIC, DMC, etc. solve this a layer higher via manifests: editing is substituting object names, deletion removal.

Permits elegant other solutions, such as merkle tries for validation of an entire resource.

Manifests

GDPR: omitting a content part isn’t the same as deleting it.

Streaming: real-time updates to manifests are Not My Problem.

Multiple authors: manifests are signed, but by whom?

Eventual consistency: multiple conflicting manifests may be “equally valid”; merging those must result in a new, improved manifest.

Vessel Extent Ordering

Goal: make manifest a result of the synchronization state, not a separate issue.

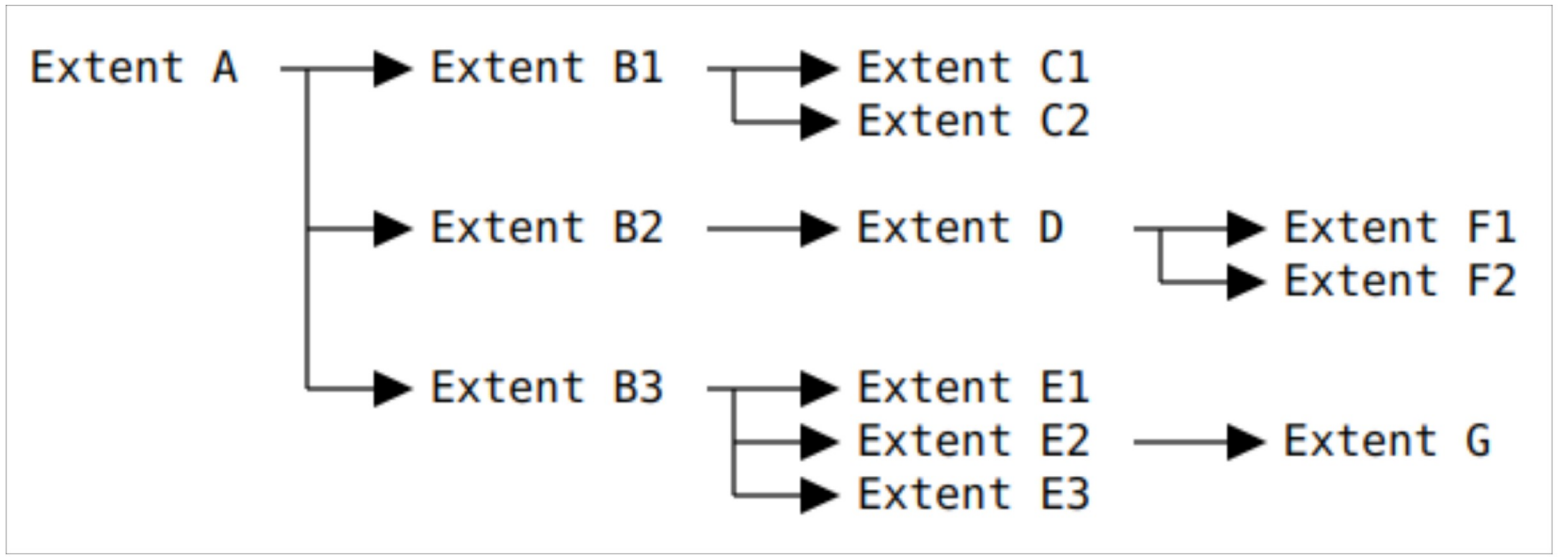
First “extent” has a random identifier.

Every subsequent extent’s identifier is hash of parent identifier & author identifier; this is verifiable, deterministic and conflict free, which permits eventually consistent approaches.

Ideal parent identifier is the leaf of the longest, “most linear” path in the tree (with tie breaker), this is deterministic for any synchronization state.

Vessel Extent Ordering

Which extent to select as parent?



Vessel Extent Ordering

No.	Path	Edge Weights	Cumulative Path Weight
1	A -> B1	1/3	0.333
2	A -> B2	1/3	0.333
3	A -> B3	1/3	0.333
4	A -> B1 -> C1	1/3 + 1/2	0.833
5	A -> B1 -> C2	1/3 + 1/2	0.833
6	A -> B2 -> D	1/3 + 1	1.333
7	A -> B1 -> D -> F1	1/3 + 1 + 1/2	**1.833**
8	A -> B1 -> D -> F2	1/3 + 1 + 1/2	**1.833**
9	A -> B3 -> E1	1/3 + 1/3	0.666
10	A -> B3 -> E2	1/3 + 1/3	0.666
11	A -> B3 -> E2 -> G	1/3 + 1/3 + 1	1.666
12	A -> B3 -> E3	1/3 + 1/3	0.666

Simple edge weight calculation is 1 divided by number of siblings.

Path weight is sum of edge weights in path.

This selects F1 or F2 as equal candidates.

Identifier ordering scheme serves as tie breaker and selects F2.

E2EE

Extents must be signed, but encryption is optional.

Signing either via public key signature or AEAD when also encrypted.

Presents key distribution challenge (as acknowledged in RFC7927).

But vessel can multiplex content, so key distribution can be multiplexed as well within the same resource.

Multiplexing

Content isn’t binary data, it’s data sections.

Each section has a type and a topic; the type specifies how to interpret a section, and the topic the context in which to interpret it.

Perfectly suitable for an AAA topic that contains e.g. sections informing consumers that an author identifier can be trusted to produce valid extents (out of scope, future draft!)

A content type section permits simple embedding of file data with sufficient meta information for clients.

Editing and Deleting

The extent identifier is permanently linked to an author, and the position of the extent within the tree is also permanent.

Extents also contain a “version” – an author-scoped counter – that authors increment whenever they create or modify an extent.

The Named Data Object “object name” is strictly speaking a tuple of extent identifier and counter.

Deletion is erasing content sections, then re-signing and updating the extent.

Right to be Forgotten

The manifest approach may include an out-of-band notification for caches to delete outdated blocks; it’s easy for a malicious cache to ignore this.

With E2EE, there is no possibility for caches to distinguish between an edit and a deletion.

It’s still possible for a cache to keep all versions of an extent, but at a cost. It’s easier to be well-behaved than to be malicious.

Limitations

The container format doesn’t solve everything about e.g. collaborative editing, but:

In a single-author use-case, it provides for improved streaming friendliness because the manifest is just the result of the next extent arriving.

In a multi-author use-case, it already provides partial ordering.

E2EE and the ability to embed AAA helps: fundamental operations should be infrastructure for the upper layers.

AAA in Vessel

Interpeer developed CAProck (cryptographic capabilities) for distributed authorization tokens – can be embedded or distributed out-of-band (ISOC foundation).

We are looking into possibly extending CAProck to also provide key exchange.

Needs separate draft for such CAProck based AAA sections in vessel.

THANK YOU

INTERPEER